LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

AD-A276 364

DTIC ACCESSION NUMBER

LEVEL

1

INVENTORY

WL-TR-94-1010

DOCUMENT IDENTIFICATION

Feb 94

Approved for public release
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR
NTIS GRA&I ☑
DTIC TRAC ☐
UNANNOUNCED ☐
JUSTIFICATION

BY

DISTRIBUTION/

AVAILABILITY CODES

DISTRIBUTION	AVAILABILITY AND/OR SPECIAL	
A-1		

DISTRIBUTION STAMP

DTIC
ELECTE
MAR 1 1994
S C D

DATE ACCESSIONED

H A N D L E W I T H C A R E

DATE RETURNED

94 4

DATE RECEIVED IN DTIC

94-06191

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

DTIC FORM JUN 90 70A

DOCUMENT PROCESSING SHEET

PREVIOUS EDITIONS MAY BE USED UNTIL STOCK IS EXHAUSTED.

LOAN DOCUMENT

WL-TR-94-1010

TECHNOLOGIES FOR AVIONICS EMBEDDED COMPUTER SYSTEMS

CHARLES P. SATTERTHWAITE

FEBRUARY 1994

FINAL REPORT FOR 02/17/93

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7409

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Charles P. Satterthwaite
Project Engineer
WL/AAAF-3

Robert L. Harris
Chief, WL/AAAF-3

Donald M. Morris
Chief, WL/AAAF

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/AAAF, WPAFB, OH 45433-6543 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

FEB 1994 FINAL 02/17/93

TECHNOLOGIES FOR AVIONICS EMBEDDED COMPUTER SYSTEMS

C
PE
PR 9994
TA 00
WU 00

CHARLES P. SATTERTHWAITE

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7409

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7409

WL-TR-94-1010

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

12

UNCLASSIFIED UNCLASSIFIED UNCLASSIFIED UL

TECHNOLOGIES FOR AVIONICS EMBEDDED COMPUTER SYSTEMS

Charles P. Satterthwaite
WL/AAAF-3
WPAFB, OH 45433-6543

INTRODUCTION

Avionics Embedded Computer Systems are an important part of the United States Air Force's plans for using advanced technologies to keep their aircraft highly capable and superior. Highly desirable features such as reconfigurability, rapid turnaround, expandability, and testability are made increasingly available through the use of these embedded systems.

This paper discusses Avionics Embedded Computer Systems primarily through a discussion of their Operational Flight Programs (OFPs), near and far term requirements of OFP maintainers, and current technologies being developed for OFP maintenance.

OVERVIEW

AVIONICS EMBEDDED COMPUTER SYSTEM (ECS) Defined

Avionics Embedded Computer Systems are computer processors which are onboard airborne weapon system platforms such as aircraft and missiles.

OPERATIONAL FLIGHT PROGRAM (OFP) Defined

An Operational Flight Program is the software program of a embedded computer system which enables that system to perform its interactive tasks as designed.

OFP's ROLE IN THE WEAPON SYSTEM

Embedded computers are increasingly called upon to provide high-tech solutions to complex multiple threat type environments for today's generation of weapon systems. The empowering of an embedded computer is through its software, which is the OFP. In understanding the role of an OFP, one must thoroughly understand the threat, the weapon system, the mission, and the embedded computer system.

MAINTENANCE OF AN OFP

OFPs are easier to maintain than mechanical units. An OFP does not break or wear out. It does not make a mistake, but it could contain faulty logic, which it follows faithfully. To maintain an OFP, one needs to be able to manipulate its sub-functions, without destroying its known integrity. There also has to be a method by which the OFP can be interactively examined and tested. Maintenance of OFPs requires a library of current and historical files which encompass the various block cycles and their unique versions as well as documentation, specialized test patch software, and any other records related to the OFP.

HOW DOES AN OFP WORK?

The Operational Flight Program (OFP) literally is the software portion of a embedded computer system. The computer and its periphery interfaces make up the system hardware. The hardware enabled by the OFP software describes the whole system.

The embedded computer system (one of many varieties of microprocessors) has partitioned memory which is filled with some type of machine level (binary) code. The OFP is loaded into this partitioned memory, and when enabled, empowers the whole system to perform its desired functions. Each embedded computer system has an instruction set which is burned into its Read Only Memory (ROM). The instruction set allows the embedded computer maintainer access and the capability to optimize the remaining partitioned memory. The level of sophistication of a embedded computer system is described by its instruction set, its memory, and its throughput.

HOW IS AN OFP CHANGED?

DIAGNOSIS/ANALYSIS/ISOLATION/INTEGRATION/TEST

Given the task of changing an OFP (making a new version or even a new block cycle), several steps are followed to bring about the change. First, the requested changes are diagnosed so that their purpose is understood. Engineers and pilots don't always view life in parallel, so careful review keeps the OFP maintainer on track. Once the OFP maintainer thoroughly understands the change request, he makes an analysis as to which OFP areas he must alter. Usually the OFP is made up of a series of modules with specialized functions which will be expanded upon later.. The OFP maintainer will next isolate these modules by making copies of them and implementing his design changes to his copies. The OFP maintainer integrates his assembled modules by linking it together with the other unaltered modules to form his own unique OFP. The OFP maintainer's final task is to test out his OFP by putting it through an acceptance test procedure, which wrings out the new OFP. For a sizable OFP with numerous change requests, several maintainers would follow these procedures simultaneously, and then a lead maintainer would integrate and test the new OFP.

THE WEAPON SYSTEM/MISSION

In order to make OFP changes, a maintainer must understand the weapon system, for which his embedded computer is a part, and the mission for which that weapon system is required for. Many times the availability for new functions in a embedded computer system are limited, so that a tradeoff analysis must be performed in order to optimize the mission and the weapon system. A sub-function which is rarely (or never) utilized might be sacrificed in order to accommodate a new requirement of higher priority.

THE SUPPORT ENVIRONMENT

In order to maintain an OFP, the maintainers require a dedicated computer system and a simulation environment. The dedicated computer system allows the maintainer to access OFPs as well as copy and alter OFPs as required. The simulation environment allows maintainers to run their OFPs allowing them to debug and test interactively.

The hardware of an dedicated computer system usually includes mainframe computers (or powerful engineering workstations), various types of printers, various disk storage devices, networking, and several access terminals. An example used by the F-15 Central Computer OFP Maintainers is the Harris Operating System with Harris 800 and 1200 Mainframes, as well as a complimentary host of Harris Printers, Disk Drives, and Reel to Reel Drives.

Some confusion can be cleared up here between an embedded computer and a dedicated computer. The embedded computer is the target processor which is part of the weapon system. A dedicated computer is outside of the weapon system and is used to support the embedded computer system.

THE DEDICATED COMPUTER SYSTEM

The dedicated computer system allows maintainers to keep copies of all the OFP's software, documentation, peculiar support utilities, and flow diagrams. It is through the dedicated computer system that a maintainer gains access to OFPs; edits or creates new versions of OFPs; updates support documentation; manages the OFP configuration; trains; and enhances the support environment with additional hardware or software.

The software of the dedicated computer system is usually peculiar to the hardware vendor. The Harris System mentioned above has its own operating language, editors, compilers, linkers, etc. The OFP maintainers also develop specialized utilities (usually in FORTRAN) to expedite and enhance the procedures they follow in the maintenance and development of OFPs.

The dedicated computer system is also comprised of system conventions which help to maintain configuration management, security regulations, and proper operation of the dedicated computer system.

SIMULATION ENVIRONMENT

OFPs must have a means by which to operate real-time, that is, loading them up in their target processor and exposing them to the range of conditions (or a reasonable subset of those conditions) they encounter when operational. This should allow the maintainer to actively debug the OFP. The degree of complexity of the OFP's environment is directly related to the complexity of this simulation environment. In the case of a typical fire control computer, you need a means to represent the full-up avionics suite and the dynamic environment the fighter encounters. You also need an interface to all cockpit controls and switches and an interface between the dedicated computer system and the simulation environment. Finally you need competent maintainers who know how to make the system work.

The simulation can range from a fully operational weapon system (flight testing is very expensive) to an all-software engineering workstation. Usually the simulation is a representative set of the weapon system's LRUs (Line Replaceable Units) with software emulating the cockpit and the dynamic environment.

Interaction with the simulation environment is through the dedicated computer system. Simulation utilities hosted on the dedicated computer system allow you to load an OFP into its target processor and exercise it dynamically or statically. These utilities also allow you to record, patch, debug, freeze, and initialize the OFP.

THE AVIONICS INTEGRATION SUPPORT FACILITY (AISF)

The facility which houses the dedicated computer system(s) and the simulation environment(s) is the Avionics Integrated Support Facility (AISF). Another name for the AISF is the Centralized Software Support Activity (CSSA). The AISF supports one or more embedded computer system and its OFP.

SHORT TERM EMBEDDED COMPUTER SYSTEM SOFTWARE SUPPORT AND MAINTENANCE REQUIREMENTS

IMPROVED TESTING METHODS

Currently, the capability to accurately and consistently test OFPs is decreasing as the OFP workload increases. Unique test cases have to be built for each area of OFP code which is upgraded. Many times the effort involved in building the test case is greater then the coding and integration efforts surrounding OFP code upgrades. Another serious problem is that overall OFP test coverage is becoming less reliable as OFPs become more complex. The acceptance test procedure is the primary method of testing OFPs. This method is very manually intensive, requiring the setting up of test cases, the changing of switches and dials, and the visual verification of controls and displays against expected values. The automation of acceptance test procedures would greatly enhance OFP Maintainers existing capabilities as well as allow them to build more complex and comprehensive test scenarios. New methods of testing are also greatly needed to augment current testing practices. Some of these new methods include graphical techniques which identify code dependencies and interrelationships; statistical techniques which would suggest reliable test sampling ; capacity evaluation, which would highlight over stressed or utility software resources; and formal methods, which would provide mathematical models of the OFP processes.

IMPROVED DOCUMENTATION METHODS

This area is critical for improving the OFP Maintenance Process. Currently most documentation is done as an after-coding effort. Unfortunately, this leaves many holes in a system which needs good traceability throughout the life-cycle of the OFP. Besides being under utilized, documentation is nonstandard, so following patches of documentation that do exist is difficult at best. Technologies exist, mainly through Computer Aided Software Engineering (CASE) tools which, if properly implemented and utilized, would allow comprehensive documentation to be part of the OFP Maintenance Process.

EXPANDED REVERSE ENGINEERING CAPABILITIES

Many times OFP maintenance and development efforts rely on past lessons learned and old versions of OFPs, their documentation, and their testing scenarios in order to update or correct change requests. Usually this practice of reverse engineering is left to the individual maintainer's best guess as to how and when to perform. It would greatly facilitate OFP maintainers to have a repeatable process in which they could archive and access all of the information related to the OFP's lifecycle. Besides being accessible, this information should be functionally arranged by module, package, type, or unit of arrangement.

ACCESS TO ADVANCE TECHNOLOGIES

Seldom is it possible to inject the most recent technologies into weapon systems without major modifications to these systems. There are at least two factors for this. The first is the age of the target weapon system. These systems can be as old as 30 years with technologies that have become obsolete several years ago. Matching rapidly changing technology to these systems is difficult at best. The second factor is that advanced technology researchers and developers have not been encouraged to develop new technology through existing systems. The incentives have been for new break throughs, and thus focused on futuristic implementations. OFP maintainers should be allowed to have some influence on capital expenditures for advanced avionics technologies so that incentives for technology transition for existing weapon systems are elevated.

LONG TERM EMBEDDED COMPUTER SYSTEM SOFTWARE SUPPORT AND MAINTENANCE REQUIREMENTS

MATURATION OF SOFTWARE ENGINEERING IN ACADEMIA

The average maintainer of OFPs does not have academic background in software engineering. Software Engineering as an undergraduate discipline is still limited to only a few universities. Computer Engineering and Computer Science are the closest disciplines to Software Engineering offered at most major universities. These disciplines are good, but need to address specific software issues before practitioners are properly prepared for maintaining OFPs. Some of these issues include insight into embedded computer systems; validation and verification of OFPs; reuse/reengineerig; and environmental parameterization.

INCREASED EMPHASIS ON DESIGN AND QUALITY

Until recently, design was not a key issue in producing software. One of the attractive features about software is that you can build and fix things quickly. This feature has drawn many software enthusiasts from many fields. Unfortunately, this feature has also brought a negative. Huge varieties of source code based on differing degrees of design and quality constraints are sometimes mixed together. A recurring theme amongst developers, implementators, and maintainers of OFP source code is that of comprehensive and quality requirements specification. In order to specify these requirements, one has to understand all of the processes associated with the OFP from original implementation through final block cycle change.

ADVANCED METHODS OF TESTING

Advanced technology testing methods will be necessary to assure the future OFPs reliability and maintainability. Plans for future OFPs include magnitude increases in size, complexity, and inter-operability. Future methods will have to be able to decompose and analyze reconfigurable OFP applications while they are running in rapidly changing environments.

INTEROPERABILITY OF EMBEDDED SYSTEMS AND THEIR SUPPORT ENVIRONMENTS

Currently, Avionics Integrated Support Facilities (AISFs) or Centralized Software Support Activities (CSSAs) provide the maintenance support environment for OFPs. These AISFs are as unique as the OFPs they support. This uniqueness causes many problems including cost of facilities, specialized expertise, and lack of portability between weapon systems. Future AISFs will be required to service any configuration of OFPs.

AVIONICS EMBEDDED COMPUTER TECHNOLOGIES

Embedded Computer Resources Support Improvement Program (ESIP) is a command-wide program to increase the logistics supportability of weapon systems. A goal of ESIP is to develop, produce, and transition new technologies and methodologies designed to reduce logistics support costs, increase weapon system support efficiency and improve response time. Within WL/AAAF, the program is structured into two main areas of endeavor: (a) the development and transition of future Embedded Computer System (ECS) support technologies and (b) the development of techniques that will allow rapid turnaround of mission critical software in a wartime environment.

ESIP AMPSE AND ASSOCIATED TECHNOLOGIES

The ESIP Ampse (advanced multipurpose support environment) prototype is a generic test environment designed to support the update and maintenance of OFPs hosted on ECS Line Replaceable Units (LRUs). The Ampse system can also support the integration testing of new weapon system avionics technologies. The Ampse concept is an alternative to traditional support environments and Avionics Integration Support Facilities (AISFs) by providing a design which is modular and expandable. The design allows for enhancements to be made to the Ampse as the weapon system it supports is enhanced. The system utilizes Commercial-Off-The-Shelf

(COTS) components and software written in Ada for improved maintainability. The Ampse design uses WL/AAAF's developed Shared Memory Advanced Real-Time Network, Version 2 (SMARTNet-2) which connects multiple microcomputers and/or SBCs in place of the single mainframe computer found throughout Air Logistics Centers (ALCs) and Air Force test facilities. The shared memory network provides real-time data transfers between processors hosting simulation models, OFP emulators, and interface software dedicated for the Out-The-Window (OTW) display, cockpit controls/displays, and embedded computer Input/Output (I/O). In addition to SMARTNet-2, other Ampse technologies (components) include the Reconfigurable Engineers Test Console (RTEC), Simulation Monitor And Control (SimMAC) Program Group, and the Distributed Ada Real-Time Executive (DARTE). Currently, the ESIP Ampse prototype is configured to F-16A/B Block 15Z1A/Z1B in support of the F-16A/B DTS task under the F-16 Ampse project.

AUTOMATED VALIDATION (AUTOVAL)

This new tool will automatically execute test cases and log errors to support the validation of OFPs. AutoVal will provide the appropriate real-time control data to the Ampse based system necessary to execute various scenarios, simulate operator actions and log discrepancies that may be found during testing. With AutoVal, complex tests can be performed in a matter of hours and the problem of human error can be avoided. An implementation of AutoVal is currently being developed to support the OFP testing of the F-16A/B Expanded Fire Control Computer (XFCC).

ESIP VIRTUAL TEST STATION (VTS)

The VTS is a test environment concept being prototyped as part of ESIP's new Common Avionics Software Support Activity (CASSA). The test environment is a completely reconfigurable system, capable of being reconfigured to support a wide variety of tests using various combinations of avionics hardware and software. Working from either a workstation or a test console, an OFP engineer will use the VTS to configure a test station with the avionics hardware and simulation software required to support a particular test. VTS building blocks will consist of workstations, test consoles, avionics computers, avionics computer interfaces, avionics computer stimulation equipment, avionics computer monitors and controllers, and general purpose single board processors. The VTS will contain a combination of models, emulators, and actual avionics hardware to provide the environment for the avionics subsystem under test and exercise it in various scenarios. The VTS is capable of performing subsystem testing using one actual avionics subsystem or integration testing using multiple actual avionics subsystems, with the remaining avionics and the external environment in either case simulated. Assuming there are sufficient resources within the VTS, the system will support multiple test stations in various configurations all operating simultaneously. Whenever a specific VTS component is not used by one test station, it will be made available for another test station. In this way, the VTS will make maximum use of the costly simulation resources and OFP engineers will have access to the required equipment and software needed to perform a test.

AVIONICS FAULT TOLERANT SOFTWARE (AFTS)

The use of fault tolerance technologies has been demonstrated in hardware applications to increase a systems reliability, survivability, and maintainability by providing the system alternatives to mission abortion when the system is stressed through battle damage or operational fatigue. Examples of alternatives include the provision of redundant reconfigurable resources and the capability to function at a degraded state. The Avionics Fault Tolerant Software (AFTS) effort is developing similar techniques to be used in ECS applications. These techniques focus on Ada code which provides a capability for fielded systems to detect, compensate, and correct software timing and data errors during execution.

MODULAR EMBEDDED COMPUTER SOFTWARE (MECS)

The MECS effort is developing design methodologies and technologies that allow for rapid cost effective creation and post deployment support of avionics software. MECS is also developing techniques for mapping software to hardware along with a framework for the collection and use of avionics software design information.

HYPERMEDIA AVIONICS SOFTWARE SUPPORT (HASS)

The HASS effort is developing capabilities which will allow ECS maintainers and users the access to rapidly evolving computer graphics, database, text, and network technologies. These technologies will allow the individual working with ECS to interactively view documentation, visual representations, reuse libraries, simulations, and other on-line resources for maintaining the ECS.

REUSABLE ADA AVIONICS SOFTWARE PACKAGES (RAASP)

The RAASP effort is developing a capability for reusing common ECS elements such as Ada packages, test routines, documentation, algorithms, data bases, and driver routines. Also under development by RAASP is a collection system (reuse library), by which common elements can be categorized and recalled.

AUTOMATIC PROGRAMMING TECHNOLOGIES FOR AVIONICS SOFTWARE (APTAS)

The APTAS effort is developing an Automatic Programming (AP) system which will provide high-level automated software specification and design capabilities for the generation of real-time avionics applications. APTAS identifies domain-specific concepts required for solving particular avionics software problems, expresses those concepts via facts and rules, captures the facts and rules in a formal specification language, and utilizes inference procedures for problem solving.

ADVANCED AVIONICS VERIFICATION AND VALIDATION (AAV&V)

The AAV&V effort is developing techniques which greatly enhance ECS developers and maintainers abilities to verify and validate their highly complex avionics software for fielded usage. Current techniques of acceptance test are very man-in-the-loop intensive and thus time intensive and increasingly incomplete. AAV&V addresses the use of graphical techniques, code dependency techniques, statistical and probability analysis techniques, and the use of formal methods to give the ECS developer a comprehensive solution to verification and validation of his software.

COMPLEXITY METRICS FOR AVIONICS SOFTWARE (CMAS)

The CMASS effort is developing the capability of measuring the complexity of avionics software size, features, and structure of its source code. With this information ECS developers and maintainers will be able to better design, test, and implement avionics software.

AVIONICS SYSTEMS PERFORMANCE MEASURE (ASPM)

The ASPM effort is developing a capability to collect and analyze performance metrics in an Integrated Avionics Environment so that performance problems can be identified and corrected. ASPM will improve techniques for the evaluation of distributed avionics software systems as well as provide its users with a more complete understanding of their system.

DOMAIN SPECIFIC SOFTWARE ARCHITECTURE (DSSA)

The DSSA effort is identifying those areas which are common in avionics software applications such as control and display routines, radar acquisition routines, and communications and navigation routines. As the common areas are identified, software architecture can be tailored for these common features which will maximize opportunities for software reuse and rapid reprogramming.

COMMON ADA RUN TIME SYSTEM (CARTS)

The CARTS effort is developing a common Ada run-time system for ECS applications. Run-time systems include a compiler's method of tasking, reconfiguring, managing memory, system initiation, as well as others. By creating a common run-time system, CARTS provides an increased opportunity for software reuse and portability between systems.

COMPUTER AIDED SOFTWARE ENGINEERING (CASE) FOR AVIONICS

Case for Avionics is an inhouse effort with the purpose of gaining first hand experience with leading CASE Tool Technologies for the purpose of supporting the complete Embedded Computer Software Lifecycle . The method for achieving this first hand experience is by first studying and comparing the leading CASE Tools; second selecting and acquiring CASE Tools which most appropriately fit the Avionics Embedded Software paradigm; and finally utilizing the selected CASE Tools while researching and developing new technologies.

Currently CASE for Avionics is to the point where most of the Tools have been acquired and are being evaluated against existing research programs. The more prominent of the Tools acquired are IDE's Software Through Pictures and Cadre's Teamwork. Currently Systran is under contract to evaluate and implement these Tools. Other Tools being examined include Statemate by i-Logix and Verilog's Tool.

RAPID TURNAROUND MISSION CRITICAL TECHNOLOGIES

HIGH SPEED AVIONICS DATA INSTRUMENTATION SYSTEM (HADIS).

The objective of this effort is to develop methods and systems capable of facilitating the software test process of embedded avionics systems. This is to be accomplished by developing an instrumentation and data display system for high-speed avionics data. This effort targeted the F-15 APG-63 radar since there was a definite need for instrumenting the radar's high-speed data for analysis. This instrumentation and display system shall enable rapid test anomaly identification and diagnosis. This developed system shall provide a foundation for enhanced OFP testing at the Warner Robins Air Logistics Center (WR-ALC), the electronic counter-countermeasures (ECCM) Advanced Radar Test Bed (ARTB), and the Radar Test Facility (RTF).

The basic technical approach is to develop a laboratory prototype testbed system that enables enhanced testing of avionics radar software for R&M. This includes a high-speed data instrumentation system which collects, records, and retrieves high-speed data continuously from the F-15 APG-63 radar facilitating data analysis. The system will have sufficient bandwidth to provide these same functions for additional radars and avionics systems.

DATA INTEGRATION & COLLECTION ENVIRONMENT (DICE).

Current and planned avionics radar systems are highly sophisticated in their capabilities and heavily dependent on Embedded Computer Systems (ECS) for their operation. Avionics radar systems containing ECS provide a high level of flexibility in both system configuration and capability. However, increased flexibility requires increased logistical support. This support includes both hardware and software maintenance. Air Force Materiel Command (AFMC) is conducting a concentrated effort to limit the rising cost of ECS support while responding to user requests for improving these systems. This effort and the ECS Support Improvement Program (ESIP), a program implemented to address the problem of developing a rapid turnaround capability for airborne avionics radar systems, have as one of their research and development objectives to develop the necessary rapid turnaround capabilities to support mission critical ECS. Rapid Turnaround (RT) is defined as correcting a system deficiency in a timely fashion through some combination of software, firmware, and/or hardware modifications. The RT concept evolved from the overall AFMC objective to decrease the time and resources required to implement operational or maintenance changes in deployed ECS.

The objective of the DICE effort is to develop a low-cost onboard airborne radar data collection prototype system utilizing the F-15 APG-63 radar as a proof-of-concept. This collected data will aid radar analysts in the analysis of a radar system's performance and operation. As a result, DICE will enhance the rapid reprogramming process of embedded computer systems software.

REFERENCES

[1] DOD-STD-2167A, Military Standard Defense System Software Development, 29 February 1988.

[2] Harris,R.L., "Laboratory Concepts in Avionics Software", IEEE/AIAA/NASA 9th Digital Avionics Systems Conference Proceedings, 15-18 October 1990, Virginia Beach VA.

[3] Harris,R.L.,Jackson,O., "Software Engineering Tools For Avionics Embedded Computer Resources", IEEE/AIAA National Aerospace Electronics Conference Proceedings, 21-25 May 1990, Dayton OH.

[4] Jones,J.V., *Engineering Design Reliability, Maintainability, Testability*, TAB Books Inc., Blue Ridge Summit, PA, 1988.

[5] Morris,D.M., "Avionics Operational Flight Program Software Supportability", IEEE/AIAA/NASA 9th Digital Avionics Systems Conference Proceedings, 15-18 October 1990, Virginia Beach VA.

[6] Satterthwaite,C.P., *Maintaining An Operational Flight Program (OFP)*, Air Force Technical Memorandum WL- TM-91-123, Wright-Patterson AFB, OH, 1991.

[7] Computer Resource Acquisition Course, "Course Materials Volume I and II", System Acquisition School, 6575TH School Squadron, Air Force Systems Command, Brooks AFB, TX, December 1991.